What is claimed is:

1. A method of initiating performance of a computation on at least one untrusted computer, the method comprising:

partitioning the computation into a plurality of computational units that are combinable to generate a result for the computation;

generating at least one distractive computational unit;

initiating execution of both the at least one distractive computational unit and at least one of the plurality of computational units on the untrusted computer to inhibit reconstitution of the computation by an untrusted party.

2. The method of claim 1, wherein the distractive computational unit comprises a computational unit generated from partitioning a second computation.

3. The method of claim 2, wherein initiating execution of both the at least one distractive computational unit and at least one of the plurality of computational units includes interleaving the at least one distractive computational unit among multiple computational units from the plurality of computational units.

4. The method of claim 2, wherein partitioning the computation uses a different algorithm than that used to partition the second computation.

5. The method of claim 1, wherein the distractive computational unit comprises a dummy computational unit.

6. The method of claim 1, wherein the distractive computational unit comprises a computational unit generated from a second partitioning of the computation.

7. The method of claim 1, further comprising initiating execution of at least one of the plurality of computational units on a second computer.

8. The method of claim 1, further comprising initiating execution of all of the plurality of computational units on the untrusted computer.

9. The method of claim 1, wherein partitioning the computation into the plurality of computational units comprises partitioning using the Chinese Remainder Theorem (CRT).

10. The method of claim 9, wherein the computation includes a plurality of arguments, and wherein partitioning the computation into the plurality of computational units comprises:

selecting a plurality of relatively prime moduli; and

generating each computational unit by performing a modulo operation on each of the plurality of arguments using one of the plurality of relatively prime moduli.

11. The method of claim 10, wherein selecting the plurality of relatively prime moduli includes selecting each modulus from a superset of relatively prime moduli, the method further comprising:

partitioning a plurality of computations into multiple computational units using different sets of moduli selected from the superset of relatively prime moduli; and

initiating execution of computational units from multiple computations on the untrusted computer.

12. The method of claim 1, further comprising:

receiving result data generated during execution of each of the plurality of computational units; and

generating a result for the computation from the result data.

13. The method of claim 1, wherein the untrusted computer is coupled to a grid computing network.

14. The method of claim 13, wherein partitioning the computation is performed by a client computer coupled to the grid computing network.

15. The method of claim13, wherein partitioning the computation is performed by a broker computer coupled to the grid computing network, the method further comprising receiving the computation from a client computer.

16. The method of claim 1, wherein partitioning the computation, generating the distractive computational unit, and initiating execution of both the distractive computational unit and the one of the plurality of computational units on the untrusted computer are performed by at least one computer coupled to the untrusted computer, the method further comprising communicating the distractive computational unit and the one of the plurality of computational units to the untrusted computer.

17. An apparatus, comprising:

at least one processor; and

program code configured to be executed by the at least one processor to initiate performance of a computation on at least one untrusted computer by partitioning the computation into a plurality of computational units that are combinable to generate a result for the computation, generating at least one distractive computational unit, and initiating execution of both the at least one distractive computational unit and at least one of the plurality of computational units on the untrusted computer to inhibit reconstitution of the computation by an untrusted party.

18. The apparatus of claim 17, wherein the distractive computational unit comprises a computational unit generated from partitioning a second computation.

19. The apparatus of claim 18, wherein the program code is configured to initiate execution of both the at least one distractive computational unit and at least one of the plurality of computational units by interleaving the at least one distractive computational unit among multiple computational units from the plurality of computational units.

20. The apparatus of claim 18, wherein the program code is configured to partition the computation using a different algorithm than that used to partition the second computation.

21. The apparatus of claim 17, wherein the distractive computational unit comprises a dummy computational unit.

22. The apparatus of claim 17, wherein the distractive computational unit comprises a computational unit generated from a second partitioning of the computation.

23. The apparatus of claim 17, wherein the program code is further configured to initiate execution of at least one of the plurality of computational units on a second computer.

24. The apparatus of claim 17, wherein the program code is further configured to initiate execution of all of the plurality of computational units on the untrusted computer.

25. The apparatus of claim 17, wherein the program code is configured to partition the computation into the plurality of computational units using the Chinese Remainder Theorem (CRT).

26. The apparatus of claim 25, wherein the computation includes a plurality of arguments, and wherein the program code is configured to partition the computation into the plurality of computational units by selecting a plurality of relatively prime moduli, and generating each computational unit by performing a modulo operation on each of the plurality of arguments using one of the plurality of relatively prime moduli.

27. The apparatus of claim 26, wherein the program code is configured to select the plurality of relatively prime moduli from a superset of relatively prime moduli, wherein the program code is further configured to partition a plurality of computations into multiple computational units using different sets of moduli selected from the superset of relatively prime moduli, and initiate execution of computational units from multiple computations on the untrusted computer.

28. The apparatus of claim 17, wherein the program code is further configured to receive result data generated during execution of each of the plurality of computational units, and generate a result for the computation from the result data.

29. The apparatus of claim 17, wherein the untrusted computer is coupled to a grid computing network.

30. The apparatus of claim 29, further comprising a client computer coupled to the grid computing network and upon which the program code resides.

31. The apparatus of claim29, further comprising a client computer coupled to the grid computing network and upon which the program code resides, wherein the program code is further configured to receive the computation from a client computer.

32. The apparatus of claim 17, wherein the program code resides on a separate computer coupled to the untrusted computer, and wherein the program code is further configured to communicate the distractive computational unit and the one of the plurality of computational units to the untrusted computer.

33. A program product, comprising:

program code configured to initiate performance of a computation on at least one untrusted computer by partitioning the computation into a plurality of computational units that are combinable to generate a result for the computation, generating at least one distractive computational unit, and initiating execution of both the at least one distractive computational unit and at least one of the plurality of computational units on the untrusted computer to inhibit reconstitution of the computation by an untrusted party; and

a computer readable signal bearing medium bearing the program code.

34. The program product of claim 33, wherein the signal bearing medium includes at least one of a recordable medium and a transmission medium.